# JDBC + SWING:

1. Create a table called “Employee” which will store employee: Name, ID, Age, Gender, Department no. (You may decide the data types ☺)
2. Then create a suitable swing based UI to enter required data to the Employee DB. Note that the user needs to insert all the five details to insert a new employee record to the database. (Insert)
3. Insert 5 records to the table using the UI.
4. In the same UI create a another internal frame to perform search/update and delete operations.
5. Enable searching with employee ID, load and display the record in an table structure. (Search)
6. Enable editing an existing record and save that to the DB. (Update)
7. Enable selected records to be deleted from the table. (Delete)

**Discussion Questions:**

1. Discuss how exceptions are handled during the process.

* **Database Connection: When establishing a connection to the database using JDBC, exceptions such as SQLException can occur if the database is unreachable, credentials are incorrect, or other connection-related issues.**
* **Database Operations: When performing CRUD operations (insert, select, update, delete) using JDBC, exceptions like SQLException can occur if there are issues with SQL queries, database constraints, or any other database-related errors.**
* **Swing Components: When handling user interactions in the Swing-based UI, exceptions like NumberFormatException can occur if the user enters invalid data, NullPointerException if components are not initialized properly, etc.**
* **To handle exceptions in the program, we use the try-catch block. Whenever a piece of code that might throw an exception is executed, it is enclosed within a try block. If an exception occurs within the try block, the corresponding catch block that matches the type of the thrown exception will be executed to handle the exception gracefully. Additionally, we can use finally block to specify cleanup code that should be executed regardless of whether an exception occurred or not.**

1. Discuss the main issues occurred during the JDBC and Swing connection process.

* **JDBC: Connection issues, such as incorrect URL, username, or password for the database. SQLException can be thrown if there are issues with the SQL queries or the structure of the database.**
* **Swing: Layout and positioning issues with Swing components. Properly arranging components in a Swing GUI can be challenging, especially when dealing with different screen sizes and resolutions.**
* **Multithreading: Swing is not thread-safe. If Swing components are accessed and updated from multiple threads, it can lead to unpredictable behavior and GUI freezes. To avoid this, Swing components should be accessed and updated only from the Event Dispatch Thread (EDT).**

1. Discuss how you can use threads in this program.

* **Database Operations: When performing database operations using JDBC, we can perform them in a separate thread to avoid blocking the GUI's Event Dispatch Thread (EDT). Long-running database queries should not be executed on the EDT to keep the UI responsive.**
* **File I/O: If the application needs to read or write data to a file, these operations can be done in a separate thread to avoid blocking the UI.**
* **Background Tasks: If the application performs any background tasks or computations, they can be executed in a separate thread to keep the main UI thread responsive.**